

## **Relatório – EP2**

O objetivo deste relatório é mostrar um resumo do que foi feito no EP2, as mudanças feitas nas classes anteriormente entregues do EP1 e listar as mudanças nos testes de aceitação.

### ***Mudanças relativas ao EP1***

- Caixa – adicionada a responsabilidade de criar alguns objetos que anteriormente eram definidos externamente (Cozinheiro, Robo e Cardapio). O caixa passou a ser responsável pelo controle dos cupons que são gerados.
- Cardápio – adição de checagem de inclusão de pratos com mesmo nome, execução da busca com filtros, listagem de subcardápios e listagem mista.
- Prato – inclusão do tempo de preparo
- Robo – Adicionado sistema de log.

## ***EP2***

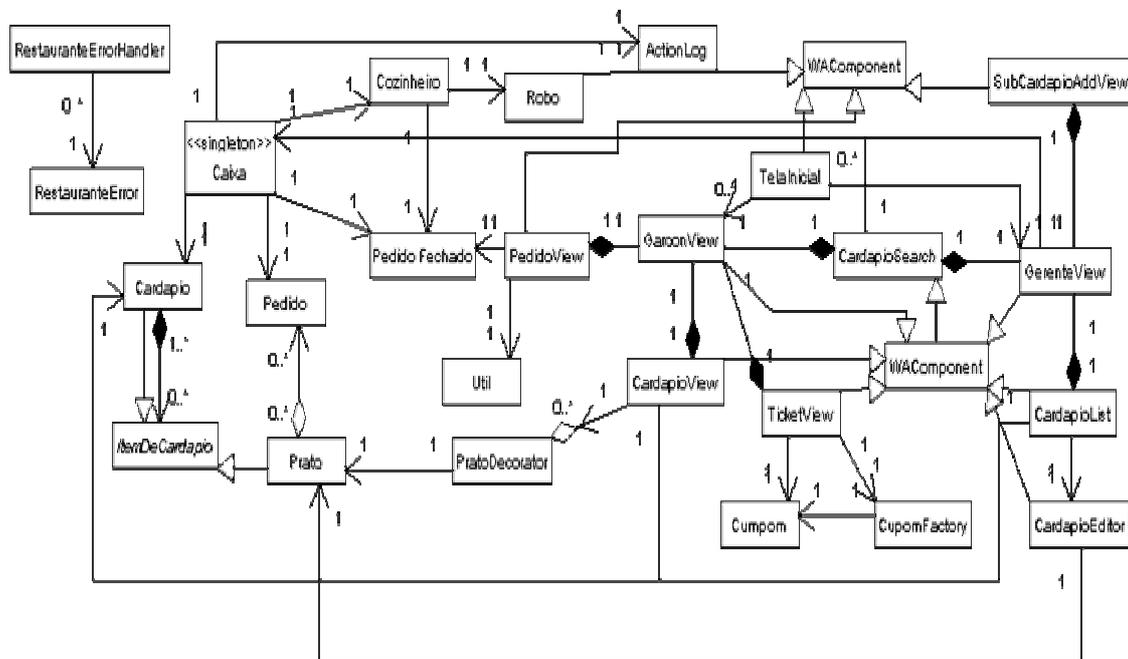
### **Instalação**

Para instalar o sistema do restaurante, é necessário configurar uma nova entrada na configuração do Seaside chamada 'restaurante', na qual o atributo 'RootComponent' deve ser direcionado para 'TelaInicial' e 'ErrorHandler' para 'RestauranteErrorHandler'. Além disso, os testes esperam que o servidor WAKomEnconded esteja rodando na porta 9090.

### **Funcionalidades extras escolhidas**

O grupo optou por escolher as funcionalidades de cupons de desconto (2 pontos), Sub Cardápios (2 pontos) e Filtro do cardápio (3 pontos) para totalizar os 7 pontos exigidos no enunciado do EP2.

## Diagrama de classes



## Classes desenvolvidas

- ActionLog – responsável por registrar informações (com horário) e também de imprimí-las na tela.
- CardapioEditor – edição dos pratos do cardápio
- CardapioList – lista os itens do cardápio
- CardapioSearch – filtro de busca dos itens do cardápio
- CardapioView – mostra o cardápio do sistema.
- Cupom – representação do cupom de descontos no sistema. Possui uma string que identifica o cupom e um valor de desconto em porcentagem.
- CupomFactory – gera um cupom de acordo com o valor do pedido. Como a regra de formação da string do cupom é composta por caracteres aleatórios concatenados com o dia e a hora em segundos, a string do cupom não deveria se repetir.
- GarconView – agrega as diferentes interfaces de atuação do garçom.
- GerenteView – agrega as diferentes interfaces de atuação do gerente
- ItemDeCardapio Classe pai da classe prato e da classe cardápio, ajuda na implementação dos subcardápios.
- PedidoView - renderiza os dados referentes ao pedido atual do caixa
- PratoDecorator - uma classe que engloba a classe prato para formatar seu valor.
- RestauranteError – Exibe as mensagens de erro do sistema de forma mais amigável.

- RestauranteErrorHandler – captura exceções para tratamento.
- SubCardápioAddView – adição e eliminação de subcardápios.
- TelaInicial – permite escolha de visualização das telas do garçom e do gerente.
- TicketView - renderiza o resultado do fechamento de um pedido, imprime o troco e o cupom gerado, se houver.
- Util – classe com funções utilitárias.

## **Testes de aceitação**

Foram feitas algumas modificações nos testes de aceitação para ficar compatível com a interface desenvolvida antes da disponibilização dos testes. As principais mudanças foram:

- a interface do cardápio oferece uma maneira diferente de se definir a quantidade de pratos de um pedido, sem haver perda de funcionalidades.
- a tela do cardápio, do pedido e do ticket foram agregadas todas na página do garçom.
- edição do cardápio: exclusão as checagens de título das páginas, inclusão do tempo de preparo, mudança no separador dos ingredientes, mensagens em geral.
- log do robo: adequação do teste à interface desenvolvida previamente.

Foram incluídos os testes de aceitação de filtro por preço (teste-filtro-maior-preco.html e teste-filtro-menor-preco.html) e filtro por ingrediente (teste-filtro-ingrediente.html e teste-filtro-not-ingrediente.html).

Foi incluído também um teste de aceitação para o cupom.

Além dos testes de aceitação, alguns testes do EP1 foram modificados e foram adicionados mais dois testes: ActionLogTest e CupomTest para o EP 2.

## **Componentes do Grupo**

Daniel Caetano - 2238318

Rafael Plana Maranzato – 5734978

Victor Lopes - 3095489